



## King's Research Portal

DOI:

[10.1007/978-3-030-30806-3\\_11](https://doi.org/10.1007/978-3-030-30806-3_11)

*Document Version*

Peer reviewed version

[Link to publication record in King's Research Portal](#)

*Citation for published version (APA):*

Hampson, C. S. (Accepted/In press). On the termination problem for counter machines with incrementing errors. In E. Filiot, R. Jungers, & I. Potapov (Eds.), *Reachability Problems - 13th International Conference, RP 2019, Proceedings* (pp. 137-148). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 11674 LNCS). [https://doi.org/10.1007/978-3-030-30806-3\\_11](https://doi.org/10.1007/978-3-030-30806-3_11)

### **Citing this paper**

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

### **General rights**

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

### **Take down policy**

If you believe that this document breaches copyright please contact [librarypure@kcl.ac.uk](mailto:librarypure@kcl.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# On the termination problem for counter machines with incrementing errors

Christopher Hampson

Department of Informatics, King's College, London, UK  
`christopher.hampson@kcl.ac.uk`

**Abstract.** In contrast to their reliable and lossy-error counterparts whose termination problems are either undecidable or non-primitive recursive, the termination problem for counter machines with *incrementing errors* is shown to be PSPACE-hard but remains solvable in EXPSpace. This is a notable decrease in complexity over that of insertion-error *channel systems* (with emptiness testing) whose termination problem is known to be non-elementary. Furthermore, by fixing the number of available counters, we obtain a tight NLOGSPACE-complete bound for the termination problem.

**Keywords:** termination · halting problem · unreliable counter machines · incrementing error · lossy error

## 1 Introduction

Reliable (Minsky) counter machines are well-known to be Turing-complete [12] and their reachability and termination problems have served as invaluable ‘master’ problems in establishing undecidable lower-bounds for a range of diverse decision problems. Furthermore, two counters are sufficient to establish Turing-completeness [12]. *Lossy counter machines (LCMs)*, by contrast, were introduced by Mayr [11] as a weakened version of Minsky’s counter machines whose counters are permitted to spontaneously ‘leak’ their contents, analogous to that of the much-studied lossy FIFO-channel systems [2,4,1,5]. Indeed, LCMs can be seen as a degenerate case of lossy channel systems (with emptiness testing) in which the channel alphabet comprises a single symbol. Mayr showed that the reachability and termination problems for LCMs are both decidable [11], with the exact complexity pinned at being ACKERMANN-complete by Schnoebelen [18] (see [17] for a comprehensive survey of non-elementary complexity classes). Indeed, just five counters are sufficient to establish non-elementary complexity, with each additional counter moving the problem further up the Fast Growing Hierarchy [18].

Less well-studied than LCMs are *incrementing counter machines (ICMs)* which are permitted to spontaneously increase the value of their counters. Incrementing errors have been considered in the context of both counter machines and their more expressive channel systems (both with and without emptiness testing) [6,4,15,13], but appear to have received far less attention than their lossy counterparts.

Insertion channel systems (without emptiness testing) were first introduced in [4], where the authors show that the termination problem (among others) is trivially decidable as every transition can be traversed with the aid of timely insertion errors. The problem thus reduces to that of cycle-finding in the underlying control-state diagram. In the presence of emptiness testing—more akin to the operational semantics of incrementing counter machines—the termination problem was shown to be TOWER-complete [3], being among the hardest problems that are primitive recursive but not solvable in elementary time [17].

With regards to the control-state reachability problem, there is no difference between lossy errors and incrementing/insertion errors for counter machines or channel systems, owing to a dualisation that reverses the ‘arrow of time’ [14,6]. Consequently, the reachability problem for both LCMs and ICMs is ACKERMANN-complete [18,6], while that of both lossy channel systems and incrementing channel systems (with emptiness testing) is HYPERACKERMANN-complete [5,14].

It appears, however, that the problem of termination for incrementing counter machines has remained unaddressed. In what follows we shall establish that the termination problem is, in general, PSPACE-hard but remains decidable in EXPSPACE. Furthermore, we show that the problem is even NLOGSPACE-complete when restricted to a fixed (finite) number of counters. Table 1 summarizes the known results relating to the termination problems for counter machines and channel systems (with emptiness testing) in the presence of lossy and incrementing errors.

	<i>Lossy</i>	<i>Incrementing</i>
Channel Systems (emptiness testing)	HYPERACKERMANN-complete [5]	TOWER-complete [3]
Counter Machines	ACKERMANN-complete [18]	PSPACE-hard in EXPSPACE Theorems 3.2 & 3.3
Counter Machines with $k$ counters	non-ELEMENTARY for $k > 5$ [18]	NLOGSPACE-complete Theorems 3.4 & 3.4

**Table 1.** Summary of termination known results for lossy and incrementing counter machines and channel systems (with emptiness testing).

## 2 Preliminaries

**Definition 2.1.** A *counter machine* is a tuple  $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$  where  $Q$  is a finite set of control-states with a designated initial state  $q_{\text{init}} \in Q$ ,  $C = \{c_1, \dots, c_n\}$  is a finite set of counters and  $\Delta \subseteq Q \times Op_C \times Q$  is a finite set of state transitions labelled with one of the following operations  $Op_C =$

$\{(c_i)^{++}, (c_i)^{-}, (c_i)^{??} : c_i \in C\}$  to *increment*, *decrement*, or *test* whether a given counter is empty.

A *configuration* of  $\mathcal{M}$  is a tuple  $(q, \mathbf{v}) \in Q \times \mathbb{N}^C$ , where  $q \in Q$  dictates the state of the machine and  $\mathbf{v} : C \rightarrow \mathbb{N}$  is  $C$ -vector describing the contents of each counter. We denote by  $\text{Conf}_{\mathcal{M}}$  the set of all possible configurations of  $\mathcal{M}$ , and define a well-quasiordering (wqo)  $\leq$  on  $\text{Conf}_{\mathcal{M}}$  by taking

$$(q, \mathbf{v}) \leq (q', \mathbf{v}') \iff q = q' \text{ and } \mathbf{v}(c_i) \leq \mathbf{v}'(c_i) \text{ for all } c_i \in C,$$

for  $(q, \mathbf{v}), (q', \mathbf{v}') \in \text{Conf}_{\mathcal{M}}$ . For each  $\alpha \in \text{Op}_C$ , we define a binary consecution relation on the configurations of  $\mathcal{M}$  by taking:

- $(q, \mathbf{v}) \xrightarrow{(c_i)^{++}} (q', \mathbf{v}')$  iff  $(q, (c_i)^{++}, q') \in \Delta$  and  $\mathbf{v}' = \mathbf{v} + \mathbf{e}_i$ ,
- $(q, \mathbf{v}) \xrightarrow{(c_i)^{-}} (q', \mathbf{v}')$  iff  $(q, (c_i)^{-}, q') \in \Delta$  and  $\mathbf{v}' = \mathbf{v} - \mathbf{e}_i$ ,
- $(q, \mathbf{v}) \xrightarrow{(c_i)^{??}} (q', \mathbf{v}')$  iff  $(q, (c_i)^{??}, q') \in \Delta$  and  $\mathbf{v} = \mathbf{v}'$  with  $\mathbf{v}(c_i) = 0$ ,

for all  $(q, \mathbf{v}), (q', \mathbf{v}') \in \text{Conf}_{\mathcal{M}}$ , where  $\mathbf{e}_i$  is the unit vector with  $\mathbf{e}_i(c_i) = 1$  and  $\mathbf{e}_i(c_j) = 0$  for  $j \neq i$ . Note that transitions of the form  $(q, (c_i)^{-}, q')$  are only enabled when  $\mathbf{v}(c_i)$  is non-zero. We write  $(q, \mathbf{v}) \xrightarrow{\mathcal{M}} (q', \mathbf{v}')$  if  $(q, \mathbf{v}) \xrightarrow{\alpha} (q', \mathbf{v}')$  for some  $\alpha \in \text{Op}_C$ . A (*reliable*) *computation* / *run* of  $\mathcal{M}$  is a sequence  $r = \langle (\sigma_{k-1}, \alpha_k, \sigma_k) \in \text{Conf}_{\mathcal{M}} \times \text{Op}_C \times \text{Conf}_{\mathcal{M}} : 0 < k < L \rangle$ , for some  $1 < L \leq \omega$ , such that  $\sigma_0 = (q_{\text{init}}, \mathbf{0})$ , where  $\mathbf{0}$  is the zero vector, and  $\sigma_{k-1} \xrightarrow{\alpha_k} \sigma_k$ , for all  $0 < k < L$ . We denote by  $\text{Runs}(\mathcal{M})$  the set of all reliable runs of  $\mathcal{M}$ .

*Lossy counter machines (LCMs)* and *incrementing counter machines (ICMs)* can be defined by way of a variation on the operational semantics of what it means for two configurations to be consecutive.

**Definition 2.2 (Lossy and Incrementing counter machines).** Given a counter machine  $\mathcal{M}$ , we define the relations  $\xrightarrow{\alpha\downarrow}$  and  $\xrightarrow{\alpha\uparrow}$  on  $\text{Conf}_{\mathcal{M}}$  by taking  $\sigma_1 \xrightarrow{\alpha\downarrow} \sigma_2$  (resp.  $\sigma_1 \xrightarrow{\alpha\uparrow} \sigma_2$ ) if and only if there are configurations  $\sigma'_1, \sigma'_2 \in \text{Conf}_{\mathcal{M}}$  such that  $\sigma_1 \geq \sigma'_1 \xrightarrow{\alpha} \sigma'_2 \geq \sigma_2$  (resp.  $\sigma_1 \leq \sigma'_1 \xrightarrow{\alpha} \sigma'_2 \leq \sigma_2$ ), which is to say that we permit the value held in the counters to spontaneously decrease (resp. increase) immediately prior to and subsequent to a reliable transition. We write  $(q, \mathbf{v}) \xrightarrow{\mathcal{M}\downarrow} (q', \mathbf{v}')$  (resp.  $(q, \mathbf{v}) \xrightarrow{\mathcal{M}\uparrow} (q', \mathbf{v}')$ ) if  $(q, \mathbf{v}) \xrightarrow{\alpha\downarrow} (q', \mathbf{v}')$  (resp.  $(q, \mathbf{v}) \xrightarrow{\alpha\uparrow} (q', \mathbf{v}')$ ) for some  $\alpha \in \text{Op}_C$ .

However, for the purposes of control-state reachability and termination, it is convenient to work with a more restrictive form of incrementing errors that encroach upon our computations only at the point of decrementing an otherwise empty counter. More precisely, we employ the following alternative definition for  $(q, \mathbf{v}) \xrightarrow{\alpha\uparrow} (q', \mathbf{v}')$ , for  $\alpha \in \text{Op}_C$ :

- $(q, \mathbf{v}) \xrightarrow{(c_i)^{++}\uparrow} (q', \mathbf{v}')$  iff  $(q, \mathbf{v}) \xrightarrow{(c_i)^{++}} (q', \mathbf{v}')$ ,
- $(q, \mathbf{v}) \xrightarrow{(c_i)^{-}\uparrow} (q', \mathbf{v}')$  iff  $(q, \mathbf{v}) \xrightarrow{(c_i)^{-}} (q', \mathbf{v}')$  or  $(q, \mathbf{v}) \xrightarrow{(c_i)^{??}} (q', \mathbf{v}')$ ,

$$- (q, \mathbf{v}) \xrightarrow{(c_i)^{??}} (q', \mathbf{v}') \text{ iff } (q, \mathbf{v}) \xrightarrow{(c_i)^{??}} (q', \mathbf{v}').$$

Such lazy ‘just-in-time’ incrementing semantics have been introduced for incrementing channel systems in [3] and used implicitly for counter machines in [6].

We define a *lossy (resp. incrementing) computation / run* of  $\mathcal{M}$  to be a sequence  $r = \langle (\sigma_{k-1}, \alpha_k, \sigma_k) \in \text{Conf}_{\mathcal{M}} \times \text{Op}_C \times \text{Conf}_{\mathcal{M}} : 0 < k < L \rangle$ , for some  $1 < L \leq \omega$ , such that  $\sigma_0 = (q_{\text{init}}, \mathbf{0})$ , where  $\mathbf{0}$  is the zero vector, and  $\sigma_{k-1} \xrightarrow{\alpha_k \downarrow} \sigma_k$  (resp.  $\sigma_{k-1} \xrightarrow{\alpha_k \uparrow} \sigma_k$ ), for all  $0 < k < L$ . We denote by  $\text{Runs}^\downarrow(\mathcal{M})$  and  $\text{Runs}^\uparrow(\mathcal{M})$  the set of all lossy and incrementing runs of  $\mathcal{M}$ , respectively.

In what follows, we will be primarily interested in the following decision problem:

ICM Termination:

*Input:* Given a counter machine  $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$ ,

*Question:* Is every incrementing run  $r \in \text{Runs}^\uparrow(\mathcal{M})$  finite?

In addition to the termination problem, we will also consider the restricted case where we admit only counter machines with a fixed number of counters.

$k$ -ICM Termination:

*Input:* Given a counter machine  $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$  such that  $|C| = k$ ,

*Question:* Is every incrementing run  $r \in \text{Runs}^\uparrow(\mathcal{M})$  finite?

As noted above, for reliable counter machines the two problems are computationally equivalent for  $k \geq 2$ , but are known to differ in complexity for lossy counter machines. We will show here that the two problems also differ in complexity for incrementing counter machines.

### 3 Results

We first show that the termination problem for ICMs is decidable in EXPSpace by establishing a doubly-exponential upper-bound on the length of all finite runs that are possible for a terminating ICM. Any incrementing counter machine exhibiting a finite run exceeding this bound must necessarily possess a non-terminating run. The termination problem can therefore be decided by a non-deterministic search for such a ‘long’ finite run which, once found, demonstrates non-termination. Such a search can be performed using at most exponential space. It then follows from Savitch’s Theorem [16] that the termination problem is decidable in EXPSpace. This is a marked contrast from the TOWER-completeness of the termination problem for incrementing channel systems (with emptiness testing) [3].

**Lemma 3.1.** *Let  $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$  be a counter machine such that every incrementing run  $r \in \text{Runs}^\uparrow(\mathcal{M})$  is finite. Then the length of each incrementing run is at most  $n^{2em!}$ , where  $n = |Q|$ ,  $m = |C|$ , and  $e$  is the base of the natural logarithm.*

*Proof.* Let  $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$  be as described above, with  $|Q| = n$  and  $|C| = m$ , and let  $r \in \text{Runs}^\uparrow(\mathcal{M})$  be any incrementing run of  $\mathcal{M}$ . The case where  $n = 1$  is trivial, so we may assume that  $n \geq 2$ . We shall refer to any sub-sequence of  $r$  as an *interval*, with its length being the number of configuration transitions it comprises. For brevity we shall refer to any transition of the form  $(\sigma, (c_i)^{??}, \sigma') \in \text{Conf}_{\mathcal{M}} \times \text{Op}_C \times \text{Conf}_{\mathcal{M}}$  as a  $c_i$ -gate and collectively as  $\Sigma$ -gates whenever  $c_i \in \Sigma$ , for  $\Sigma \subseteq C$ . An interval will be described as *gate-free* whenever it contains no  $C$ -gates.

To facilitate the proof, we define an increasing function  $T : \mathbb{N} \rightarrow \mathbb{N}$  recursively by taking

$$T(0) = 1 \quad \text{and} \quad T(k) = kT(k-1) + 2 \quad (\dagger)$$

for all  $k > 0$ . It follows from a straightforward induction that

$$T(k) = k! \left( \frac{1}{0!} + \frac{2}{1!} + \cdots + \frac{2}{(k-1)!} + \frac{2}{k!} \right) < 2k! \sum_{t=0}^{\infty} \frac{1}{t!} = 2ek!$$

for all  $k \geq 0$ , where  $e$  is the base of the natural logarithm.

For each subset  $\Sigma \subseteq C$ , let  $\chi^r(\Sigma)$  denote the length of the longest interval in which the only gates traversed belong to  $\Sigma$ . We prove by induction on the size of  $\Sigma$  that

$$\chi^r(\Sigma) < n^{T(|\Sigma|)}.$$

for all subsets  $\Sigma \subseteq C$ .

- *Base Case*) For the case where  $|\Sigma| = 0$ , we note that  $\chi^r(\emptyset) < n$  since otherwise, by the pigeonhole principle, there would be some gate-free interval  $I$  in which the same control-state appears twice. We could then construct a non-terminating run by traversing the resulting loop indefinitely, as every underlying state transition of the form  $(q, (c_i)^{++}, q')$  or  $(q, (c_i)^{--}, q')$  can always be traversed. This contradicts our assumption that every incrementing computation of  $\mathcal{M}$  is terminating.
- *Inductive Case*) Suppose that the claim holds for all subsets of size  $\leq k$  and that  $|\Sigma| = (k+1)$ . Suppose to the contrary that  $\chi^r(\Sigma) \geq n^{T(k+1)}$  and let  $I = \langle (\sigma_t, \alpha_t, \sigma_{t+1}) : t < \chi \rangle$ , be such an interval of length  $\chi = \chi^r(\Sigma)$ . It follows from that induction hypothesis that  $I$  contains at least one  $c_j$ -gate, for each  $c_j \in \Sigma$ . Choose  $c_i \in \Sigma$  and partition  $I$  into subintervals  $I_1, \dots, I_s$  by abscising all  $c_i$ -gates, as illustrated in Figure 1. Note that we can abscise at most  $n$  consecutive  $c_i$ -gates between each interval, else by the same argument as above, we could construct a non-terminating run by traversing a loop of such gates indefinitely.

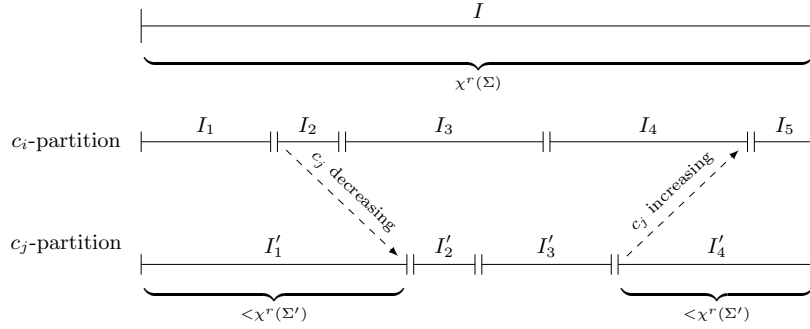
The resulting subintervals only contain  $\Sigma'$ -gates, where  $\Sigma' = \Sigma - \{c_i\}$ . Hence, by the induction hypothesis, the length of each subinterval can be at most  $\chi^r(\Sigma') < n^{T(k)}$ , since  $|\Sigma'| = k$ . It follows that

$$|I| \leq s \cdot \chi^r(\Sigma') + (s+1)n$$

which is to say that

$$s \geq \frac{|I| - n}{\chi^r(\Sigma') + n} > \frac{n^{T(k+1)} - n}{n^{T(k)} + n} \geq \frac{1}{2} \cdot \frac{n^{T(k+1)-1}}{n^{T(k)-1}} \geq n^{T(k+1)-T(k)-1}$$

for  $n \geq 2$ . It then follows from (†) that  $s > n^{kT(k)+1}$ .



**Fig. 1.** Illustration of two partitionings of  $I$  into subintervals by abscising all  $c_i$ -gates and all  $c_j$ -gates, respectively.

For each subinterval  $I_\ell$ , let  $\text{start}(\ell) = (q, v \upharpoonright \Sigma')$  denote the configuration at the start of  $I_\ell$  restricted to only those counters occurring in  $\Sigma'$ .

Note that for each  $c_j \in \Sigma'$ , the first transition of  $I_\ell$  must appear in some interval  $I'$  in which no  $c_j$ -gate appears, or else must be itself a  $c_j$ -gate. In the latter case, we require that the value of  $v(c_j)$  at  $\text{start}(\ell)$  is zero. In the former case, either the start of  $I'$  or the end of  $I'$  is contained within  $I$ , else  $I$  would not contain any  $c_j$ -gates.

- If the start of  $I'$  is contained in  $I$  then the value of  $v(c_j)$  at the start of  $\text{start}(\ell)$  can be at most  $\chi^r(\Sigma')$  since otherwise the counter could not have been incremented (using the lazy semantics) since being emptied to traverse the gate at the start of  $I'$ .
- Alternatively, if the end of  $I'$  is contained in  $I$  then the value of  $v(c_j)$  at  $\text{start}(\ell)$  can be at most  $\chi^r(\Sigma')$  since otherwise the counter could not be depleted in time to traverse the gate at the end of  $I'$ .

Hence, for each  $\ell = 1, \dots, s$ , there are at most  $n$  choices for the state of  $\text{start}(\ell)$  and  $\chi^r(\Sigma') < n^{T(k)}$  choices for the values of each of the counters

$c_j \in \Sigma'$  in  $\text{start}(\ell)$ . This gives at most  $n \cdot (n^{T(k)})^k = n^{kT(k)+1}$  possible choices for  $\text{start}(\ell)$ . However, since  $s > n^{kT(k)+1}$ , by the pigeonhole principle there must be at least two intervals  $I_\ell$  and  $I_{\ell'}$  such that  $\text{start}(\ell) = \text{start}(\ell')$ , where  $1 \leq \ell < \ell' \leq s$ .

Consequently, it is possible to construct a non-terminating run by traversing the resulting loop indefinitely, as the two partial states agree on all counters from  $\Sigma'$ , and the only  $c_j$ -gates to be traversed are those from  $\Sigma'$ . Any counters from  $(C - \Sigma')$  are free to be incremented or decremented without impeding the computation. Again, this contradicts our assumption that every incrementing run  $r' \in \text{Runs}^\uparrow(\mathcal{M})$  is terminating. Hence, by contradiction, we must have that  $\chi^r(\Sigma) < n^{T(k+1)}$ .

Hence, it follows that  $\chi^r(\Sigma) < n^{T(|\Sigma|)}$  for all  $\Sigma \subseteq C$ . In particular, we note that the maximum length of  $r$  is given by  $\chi^r(C) < n^{\overline{T(m)}} < n^{2em!}$ , as required.  $\square$

With this upper-bound placed on the maximum possible length of runs for terminating ICMs, we are able to secure an EXPSPACE upper-bound on the complexity of the termination problem.

**Theorem 3.2.** *The ICM Termination problem is decidable in EXPSPACE.*

*Proof.* By Lemma 3.1, it is sufficient to identify whether a given counter machine has a finite run whose length exceeds  $n^{2em!}$ , where  $n = |Q|$  and  $m = |C|$ . This can be achieved via a non-deterministic search using at most exponential space, by storing only the current length and final configuration of the run as the search progresses. Both the length and the final configuration can be encoded as binary strings requiring at most  $O(\log_2(n)m!)$  bits of data, which is at most exponential in  $m$  and logarithmic in  $n$ . Should a run exceeding the aforementioned bound be found, we may conclude that the counter machine has a non-terminating incrementing run.

Hence, the *non*-termination problem for incrementing counter machines is decidable in NEXPSPACE, and so it follows that both the termination and non-termination problems belong to EXPSPACE, as required.  $\square$

This result stands in marked contrast to the lofty ACKERMANN-completeness of the termination problem for *lossy* counter machines [18], despite the equivalence of the reachability problem for the two types of unreliable machines [13]. Moreover, this result also highlights a jump in complexity from the relatively modest EXPSPACE for incrementing counter machines to the non-ELEMENTARY complexity for incrementing *channel systems* (with emptiness testing) [3].

Next, we will provide a lower-bound on the complexity of the termination problem for ICMs by showing that an incrementing counter machine with  $m = |C|$  counters is capable of simulating a run of a *reliable* counter machine whose counters are bounded by  $2^{\lfloor m/2 \rfloor}$ . This, in turn, provides us with a mechanism by which we can simulate any Turing machine that operates in space bounded by  $\lfloor m/2 \rfloor$ , thereby providing us with a PSPACE-hard lower-bound for the termination problem for ICMs.



**Theorem 3.3.** *The ICM Termination problem is PSPACE-hard.*

*Proof.* Let  $X \subseteq \{0, 1\}^*$  be an arbitrary problem solvable in PSPACE, which is to say that there is some Turing machine  $\mathcal{T}_X$  and polynomial function  $p(n)$  such that  $\mathcal{T}_X$  terminates on all inputs and accepts  $w \in \{0, 1\}^*$  if and only if  $w \in X$ , using at most  $p(|w|)$  tape cells. Following Minsky [12], we may translate  $\mathcal{T}_X$  together with a given input word  $w \in \{0, 1\}^*$  into a reliable counter machine  $\mathcal{M}_w^X = \langle Q, C, q_{\text{init}}, \Delta \rangle$ —polynomial in the size of  $w$  and constructible in polynomially time—such that  $w$  is accepted by  $\mathcal{T}_X$  if and only if  $\mathcal{M}_w^X$  has a reliable run that reaches some accepting state  $q_{\text{accept}} \in Q$ . Moreover, the value of the counters of  $\mathcal{M}_w^X$  never exceeds  $2^N - 1$ , where  $N = p(|w|)$  is the maximum length of tape required by  $\mathcal{T}_X$  on input  $w$ . We may modify  $\mathcal{M}_w^X$  by adding a looping transition to  $q_{\text{accept}}$  so that  $\mathcal{M}_w^X$  has a non-terminating run if and only if  $w$  is accepted by  $\mathcal{T}_X$ .

We may then construct an ICM  $\mathcal{M}' = \langle Q', C', q'_{\text{init}}, \Delta' \rangle$ , polynomial in the size of both  $\mathcal{M}_w^X$  and  $N$ , such that  $\mathcal{M}'$  has a non-terminating incrementing run  $r' \in \text{Runs}^\dagger(\mathcal{M}')$  if and only if  $\mathcal{M}_w^X$  has a non-terminating reliable run  $r \in \text{Runs}(\mathcal{M}_w^X)$ , along which the counters are bounded by  $2^N$ . To achieve this, we first designate counters  $c_i^0, \dots, c_i^{N-1} \in C'$ , for each  $c_i \in C$ , so that the value of counter  $c_i$  for a given valuation  $\mathbf{v} : C \rightarrow \mathbb{N}$  can be represented in binary as

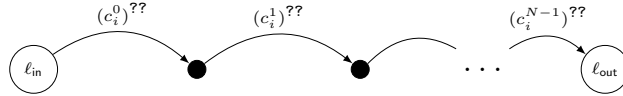
$$\theta_{\mathbf{v}}(c_i) = \sum_{j=0}^{N-1} 2^j \min\{1, \mathbf{v}(c_i^j)\}$$

In other words, the emptiness (0) or non-emptiness (1) of each of the counters  $c_i^j$  collectively represent the value of  $c_i$  in binary.

We also require a second copy  $\bar{c}_i^0, \dots, \bar{c}_i^{N-1} \in C'$ , for each  $c_i \in C$ , so that any incrementing errors can be detected and the computation terminated as a result. To achieve this we shall enforce that any increment (resp. decrement) to  $c_i^j$  is followed by a decrement (resp. increment) to  $\bar{c}_i^j$  so that, over reliable runs, the pair  $(c_i^j, \bar{c}_i^j)$  acts like a binary switch with exactly one of the counters being empty at any given time.

For each  $(\ell, \alpha, \ell') \in \Delta$ , we construct a circuit of transitions that emulates the effect of  $\alpha$  on the corresponding value of  $\theta$ .

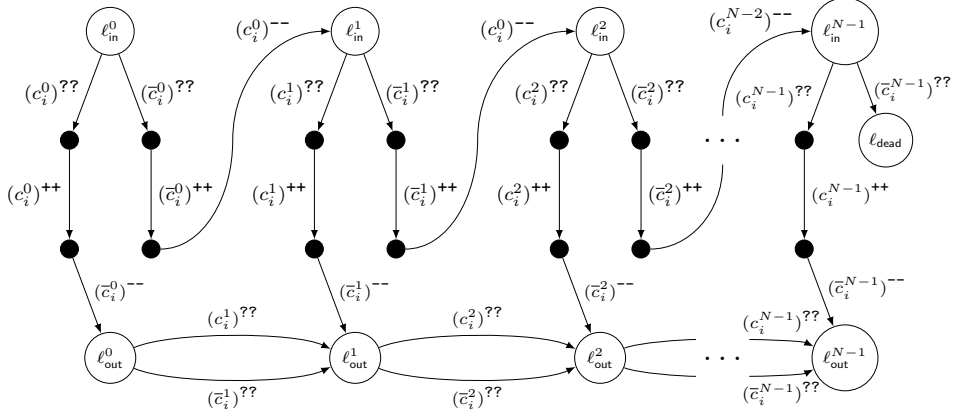
- *Case  $\alpha = (c_i)^{??}$ :* We can check whether  $\theta_{\mathbf{v}}(c_i) = 0$  by a series of emptiness checks to confirm that each of the counters  $c_i^j$  are empty, for  $j < N$ , as illustrated in Figure 2.



**Fig. 2.** Circuit emulating the operation  $(c_i)^{??}$ .

It is straightforward to check that there is an incrementing path  $(\ell_{\text{in}}, \mathbf{v}) \xrightarrow{\mathcal{M}^\uparrow^*} (\ell_{\text{out}}, \mathbf{v}')$  if and only if  $\theta_{\mathbf{v}}(c_i) = 0$ .

- *Case  $\alpha = (c_i)^{++}$ :* To increment the value of  $\theta_{\mathbf{v}}(c_i)$  by one, we can execute the circuit illustrated in Figure 3.



**Fig. 3.** Circuit emulating the operation  $(c_i)^{++}$ .

A successful computation through this circuit from  $\ell_{\text{in}}^0$  to  $\ell_{\text{out}}^{N-1}$  simulates standard binary addition by one by ‘resetting’ each  $c_i$  counters to zero in turn until the first empty  $c_i$  counter is found, which is then set to one, resetting  $\bar{c}_i$  to zero in the process. Any remaining counters that have not been inspected are then checked to ensure that exactly one of  $c_i$  and  $\bar{c}_i$  are non-zero. It follows that there is an incrementing path  $(\ell_{\text{in}}^0, \mathbf{v}) \xrightarrow{\mathcal{M}^\uparrow^*} (\ell_{\text{out}}^{N-1}, \mathbf{v}')$  if and only if:

- (i)  $\mathbf{v}(c_i^j) + \mathbf{v}(\bar{c}_i^j) = 1$ , for all  $j < N$ , and
- (ii) If  $\mathbf{v}'(c_i^j) + \mathbf{v}'(\bar{c}_i^j) = 1$  then  $\theta_{\mathbf{v}'}(c_i) = \theta_{\mathbf{v}}(c_i) + 1$ .

Note that in the case where  $\theta_{\mathbf{v}}(c_i) = 2^N - 1$  (i.e. the counter is full) it is not possible to reach  $\ell_{\text{out}}^{N-1}$  and instead we terminate in a dead-end state  $\ell_{\text{dead}}$ .

- *Case  $\alpha = (c_i)^{-}$ :* To decrement the value of  $\theta_{\mathbf{v}}(c_i)$  by one, we can use an analogous set of a transitions, but with the roles of  $c_i^j$  and  $\bar{c}_i^j$  exchanged, for  $j < N$ . The resulting circuit simulates binary subtraction by one and then ensures that exactly one of  $c_i$  and  $\bar{c}_i$  are non-zero. It is similarly straightforward to check that there is an incrementing path  $(\ell_{\text{in}}^0, \mathbf{v}) \xrightarrow{\mathcal{M}^\uparrow^*} (\ell_{\text{out}}^{N-1}, \mathbf{v}')$  in the resulting circuit if and only if:

- (i)  $\mathbf{v}(c_i^j) + \mathbf{v}(\bar{c}_i^j) = 1$ , for all  $j < N$ , and
- (ii) If  $\mathbf{v}'(c_i^j) + \mathbf{v}'(\bar{c}_i^j) = 1$  then  $\theta_{\mathbf{v}'}(c_i) = \theta_{\mathbf{v}}(c_i) - 1$ .

Similarly, in the case where  $\theta_v(c_i) = 0$  it is not possible to reach  $\ell_{\text{out}}^{N-1}$  and we terminate in state  $\ell_{\text{dead}}$ .

Note that each of the circuits are cycle-free and so do not allow for non-terminating computations to arise *within* the individual circuit. It follows that we can construct an equivalent ICM  $\mathcal{M}'$  by replacing each of the transitions of  $\mathcal{M}_w^X$  with a copy of the appropriate circuit described above, each comprising at most  $8N$  transitions. The resulting machine is at most polynomial in the size of  $\mathcal{M}_w^X$  and  $N$ , with  $|Q'| \leq 8N \cdot |\Delta| \leq 8Nn^2$  and  $|C'| = 2Nm$ , and has a non-terminating incrementing run if and only if  $\mathcal{M}_w^X$  has a reliable non-terminating run with counters bounded by  $2^N$ . We could, as well, introduce a sequence of transitions to the initial state that first increment each of the  $\bar{c}_i$  variables by one so that  $v(c_i) + v(\bar{c}_i) = 1$  at the start of the run. However, this is not required as this can be achieved with a timely incrementing error, without which any computation would quickly terminate.

As  $\mathcal{M}_w^X$  is at most polynomial in the size of  $w$  and constructible in polynomial time, it follows that  $X$  is polynomially reducible to ICM Termination, thereby demonstrating the problem to be PSPACE-hard, as required.  $\square$

Note that the above reduction requires an unbounded supply of counters for ever larger values of  $N$ . Indeed, such a reduction is not possible using only a fixed number of counters. Taking a closer look at the bound given in Lemma 3.1, we note that it is chiefly the number of counters  $m$  that is responsible for the doubly-exponential bound on the length of the incrementing runs. By fixing the number of counters, we obtain a far more tractable bound with little additional overhead.

**Theorem 3.4.** *The  $k$ -ICM Termination problem is NLOGSPACE-complete.*

*Proof.* The lower-bound is trivial and follows from a straightforward logspace reduction from the NLOGSPACE-hard reachability problem for directed graphs [10]. For the upper-bound, the proof is the same as that of Theorem 3.2, noting that for a fixed number of counters the bound given in Lemma 3.1 is logarithmic in the number of states. This, therefore, gives us an NLOGSPACE upper-bound for the *non*-termination problem. However, by the Immerman–Szelepcsényi theorem [9,19] we have that NLOGSPACE is closed under complements, thereby completing the proof.  $\square$

## 4 Discussion

1. The main problem left open by this present work is to establish where lies the exact complexity of the termination problem for incrementing counter machines. Using the same principle as in Theorem 3.3, it is not hard to construct a *terminating* ICM with exponentially long runs; for example, by connecting the state  $\ell_{\text{out}}^{N-1}$  back to state  $\ell_{\text{in}}^0$  in Figure 3. The resulting circuit contains no non-terminating computations, but is permitted to cycle through all binary representations from zero to  $2^N - 1$  before terminating in state  $\ell_{\text{dead}}$ . Unfortunately, it remains unclear whether it is possible to construct terminating

ICMs which have *doubly*-exponentially long runs that would be required for the EXPSpace upper-bound given in Theorem 3.2 to be tight.

2. Reductions from various counter machine reachability problems have been used to establish lower-bounds for several first-order modal and temporal logics endowed with additional counting quantifiers [8]. Their lossy and incrementing counterparts arise naturally in this context when we consider first-order modal logics with decreasing or expanding domains, respectively [8,7]. In particular, the recurrence problem for ICMs can be reduced to the satisfiability problem for the one-variable fragment of Linear Temporal Logic (LTL) over expanding domains with both future and next-time operators, thereby providing a  $\Sigma_1^0$ -hard lower-bound. Finite satisfiability, though decidable, can be shown to be ACKERMANN-hard by a reduction from the ICM reachability problem; this remains true even in the absence of the ‘next-time’ operator. For the fragment having a single future operator, Theorem 3.3 can be utilized to provide a PSPACE-lower bound for the satisfiability problem<sup>1</sup>. However, it is reasonable to suspect that the exact complexity may be much higher and, indeed, the decidability of this fragment still remains open.

**Acknowledgements** I would like to thank the anonymous referees for their helpful and invaluable suggestions.

## References

1. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design* **25**(1), 39–65 (2004)
2. Abdulla, P.A., Jonsson, B.: Verifying programs with unreliable channels. *information and computation* **127**(2), 91–101 (1996)
3. Bouyer, P., Markey, N., Ouaknine, J., Schnoebelen, P., Worrell, J.: On termination and invariance for faulty channel machines. *Formal Aspects of Computing* **24**(4), 595–607 (Jul 2012)
4. Cécé, G., Finkel, A., Iyer, S.P.: Unreliable channels are easier to verify than perfect channels. *Information and Computation* **124**(1), 20–31 (1996)
5. Chambart, P., Schnoebelen, P.: The ordinal recursive complexity of lossy channel systems. In: 2008 23rd Annual IEEE Symposium on Logic in Computer Science. pp. 205–216. IEEE (2008)
6. Demri, S., Lazić, R.: LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic (TOCL)* **10**(3), 16 (2009)
7. Hampson, C.: Two-dimensional modal logics with difference relations. Ph.D. thesis, King’s College London (2016)

<sup>1</sup> It is incorrectly claimed in [7] that the satisfiability problem for this logic is non-elementary, erroneously stating that the termination problem for ICMs matches that of incrementing channel systems with emptiness testing, as refuted here in Theorem 3.2.

8. Hampson, C., Kurucz, A.: Undecidable propositional bimodal logics and one-variable first-order linear temporal logics with counting. *ACM Transactions on Computational Logic (TOCL)* **16**(3), 27:1–27:36 (2015)
9. Immerman, N.: Nondeterministic space is closed under complementation. *SIAM Journal on computing* **17**(5), 935–938 (1988)
10. Jones, N.D.: Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences* **11**(1), 68–85 (1975)
11. Mayr, R.: Undecidable problems in unreliable computations. *Theoretical Computer Science* **297**(1-3), 337–354 (2003)
12. Minsky, M.L.: *Computation*. Prentice-Hall Englewood Cliffs (1967)
13. Ouaknine, J., Worrell, J.: On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science* **3**(1) (2007)
14. Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic. In: 20th Annual IEEE Symposium on Logic in Computer Science (LICS’05). pp. 188–197. IEEE (2005)
15. Ouaknine, J., Worrell, J.: On metric temporal logic and faulty turing machines. In: International Conference on Foundations of Software Science and Computation Structures. pp. 217–230. Springer (2006)
16. Savitch, W.J.: Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences* **4**(2), 177–192 (1970)
17. Schmitz, S.: Complexity hierarchies beyond elementary. *ACM Transactions on Computation Theory (TOCT)* **8**(1), 3:1–3:36 (2016)
18. Schnoebelen, P.: Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In: International Symposium on Mathematical Foundations of Computer Science. pp. 616–628. Springer (2010)
19. Szelepcsényi, R.: The method of forced enumeration for nondeterministic automata. *Acta Informatica* **26**(3), 279–284 (1988)